	Type	L #	Hits	Search Text	DBs	Time Stamp
1	BRS	L1	844783	(configure or configured or configuring or configuration or install or installed or installing or installation or initialize or initialized or initializing or initialization or load or loading or reload or reloading or download or downloading or reconfigure or reconfigured or reconfiguring or reconfiguration or reinstall or reinstalled or reinstalling or reinstallation or reinitialize or reinitialized or reinitializing or reinitialization) near5 (software or application or program or feature or device or computer)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2005/02/01 21:44
2	BRS	L2	27404	1 near5 (remote or remotely or center or central or centrally)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2005/02/01 21:45
3	BRS	L3	5674	2 near5 (communication or line or link or channel or web or www or net or lan or wan or internet or network)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2005/02/01 21:46

	Type	L #	Hits	Search Text	DBs	Time Stamp
4	BRS	L4	256538	(configure or configured or configuring or configuration or install or installed or installing or installation or initialize or initialized or initializing or initialization or load or loading or reload or reloading or download or downloading or reconfigure or reconfigured or reconfiguring or reconfiguration or reinstall or reinstalled or reinstalling or reinstallation or reinitialize or reinitialized or reinitializing or reinitialization) near5 (authorize or authorized or authorization or authenticate or authenticated or authenticating or authentication or verify or verified or verifying or verification or enable or enabled or enabling or permit or permitted or permission)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2005/02/01 21:46
5	BRS	L5	1870	2 and 3 and 4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2005/02/01 21:47
6	BRS	L6	25050	1 near10 4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2005/02/01 21:48

	Type	L #	Hits	Search Text	DBs	Time Stamp
7	BRS	L7	912	5 and 6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2005/02/01 21:48
8	BRS	L8	468	7 and (frank or franking or mail or mailing or ship or shipping or postage or meter or metering)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2005/02/01 21:51
9	BRS	L9	84	8 and (@pd<="19980615" or (@pd>="19980615" and (@ad<="19980615" or @prad<="19980615"))) Scanned Ti,Ab,kwic all	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2005/02/01 21:52
10	BRS	L10	16	("4785417" or "4812994" or "5077660" or "5107455" or "5206812" or "5508933").pn. or ((@pd<="19710101" not @pd<="19470101") and (705/401 or 705/408 or 705/410 or 713/1 or 713/2 or 713/100 or 717/168 or 717/170 or 717/171 or 717/172 or 717/174 or 717/176 or 717/177).ccls.) Scanned Ti all	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	2005/02/01 22:04

	Document ID	Issue Date	Inventor	Current OR	Current XRef	Pages
1	US 5388211 A	19950207	Hornbuckle; Gary D.	717/178	702/186; 705/32; 705/52; 705/55; 705/77; 713/187; 713/190; 713/200	23
2	US 5497479 A	19960305	Hornbuckle; Gary D.	463/29	463/42; 711/164	25
3	US 5613089 A	19970318	Hornbuckle; Gary D.	711/164		25
4	US 5649187 A	19970715	Hornbuckle; Gary D.	707/10	709/224; 709/229; 713/100; 713/2; 714/18; 717/127; 717/178	21
5	US 5734831 A	19980331	Sanders; James B.	709/223		16
6	US 5838907 A	19981117	Hansen; Peter A.	709/220	709/221; 709/223; 709/224; 709/225	34
7	US 6012100 A	20000104	Frailong; Jean-Marc et al.	709/250	709/220; 709/223	28
8	US 6067582 A	20000523	Smith; Benjamin Hewitt et al.	710/5	709/203; 717/177	14

L 9 results

	Document ID	Issue Date	Inventor	Current OR	Current XRef	Pages
9	US 6108420 A	20000822	Larose; Gordon Edward et al.	705/59	380/30	20
10	US 6195694 B1	20010227	Chen; Shuang et al.	709/220	709/203; 709/204; 709/219; 709/221	29

L9 results

	Document ID	Issue Date	Inventor	Current OR	Current XRef	Pages
1	US 5508933 A	19960416	Abumehdi; Cyrus	705/408		7
2	US 5206812 A	19930427	Abumehdi; Cyrus	705/410	710/100	12
3	US 5107455 A	19920421	Haines; John G. et al.	710/8	705/60	12
4	US 5077660 A	19911231	Haines; John G. et al.	705/410		17
5	US 4812994 A	19890314	Taylor; Michael P. et al.	705/410	340/5.28; 340/5.42	11
6	US 4785417 A	19881115	Obrea; Liana	705/410	714/38	6

L10 results

US-PAT-NO: 5388211
DOCUMENT-IDENTIFIER: US 5388211 A
TITLE: Method and apparatus for remotely controlling and monitoring the use of computer software
DATE-ISSUED: February 7, 1995
INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Hornbuckle; Gary D.	Pebble Beach	CA	N/A	N/A

US-CL-CURRENT: 717/178, 702/186 , 705/32 , 705/52 , 705/55 , 705/77 , 713/187 , 713/190 , 713/200

ABSTRACT: Remote control of the use of computer data is described in a system for renting computer software which derives use and billing information, prevents unauthorized use, maintains integrity of the software and controls related intercomputer communications. A user at a target computer "downloads" programs or data, via a telephone line and remote control modules, from a host computer. Usage of the programs or data by the target computer or other accounting data are recorded and stored and, at predetermined times, the host computer "uploads" the usage data for processing. Other features include: (1) software and usage security for rental programs; (2) a polynomial generator/checker for generating block check characters for assuring integrity of data transmitted and received; (3) a voice-data switch for switching between data communication and normal telephone communication; and (4) an audio amplifier and speaker for monitoring of activity on the communication line during data transfers.
32 Claims, 13 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 11

---------- KWIC ---------

Abstract Text - ABTX (1): Remote control of the use of computer data is described in a system for renting computer software which derives use and billing information, prevents unauthorized use, maintains integrity of the software and controls related intercomputer communications. A user at a target computer "downloads" programs or data, via a telephone line and remote control modules, from a host computer. Usage of the programs or data by the target computer or other accounting data are recorded and stored and, at predetermined times, the host computer "uploads" the usage data for processing. Other features include: (1) software and usage security for rental programs; (2) a polynomial generator/checker for generating block check characters for assuring integrity of data transmitted and received; (3) a voice-data switch for switching between data communication and normal telephone communication; and (4) an audio amplifier and speaker for monitoring of activity on the communication line during data transfers.

DATE ISSUED - PD (1): 19950207

Brief Summary Text - BSTX (3): For purposes of the present invention, rental computer software refers to the service of providing computer software to customers

(hereafter also users) on a pay-as-used basis, where the software is executed on the customer's own personal computer. In the past, the only software offered for "rent" was software installed on centrally located computers, accessible via remotely located workstations or terminals. Such systems are well-known as "time-sharing" systems.

Brief Summary Text - BSTX (7): In the relevant prior art, U.S. Pat. No. 4,361,851 discloses a television usage monitoring system comprising a modified program selector (installed in the home of a subscriber) which is used to select television programs for viewing while, at the same time, providing the selection information to a remote monitoring unit (also installed in the subscriber's home). The remote monitoring unit is connected to the subscriber's telephone line and is programmed to periodically communicate, via telephone lines, with a central computer for the purpose of transmitting the television usage data thereto. The disclosed remote monitoring system can be utilized for "[a]ccess to centralized public database networks" (see column 2, line 4). The system is also described as having the capability of producing a "disable" signal from the central computer to the remote unit if, for example, the subscriber has not timely paid charges due on his account. It should be noted that U.S. Pat. No. 4,361,851 does not disclose a system for 1) secure and remotely controlled downloading and use of computer programs and data; 2) remotely controllable monitoring of use and security of the downloaded programs and data; and 3) accessing and retrieving stored usage data. In addition, neither means for generating block check characters for data transmitted and received, nor voice-data switching capability is described.

Brief Summary Text - BSTX (11): The control module used in the proposed software rental system performs many more functions than its counterpart in the pay-for-view television system. For example, it controls and verifies that use of a program is authorized; it records the actual time that the program is used; and it protects the rental program from theft, copying, vandalism or modification. In addition, facilities for communication via the telephone lines between the control module installed at the user's site and the central or host computer are provided.

Brief Summary Text - BSTX (12): A software rental system according to the present invention is also efficient and highly automated, for performing a number of overhead functions. At the same time, in order to maximize customer satisfaction, the overhead activities of the control module are essentially transparent to the user. Thus, for example, accounting and billing activities are automated to avoid the need for manual "meter readers", and other control operations conventionally involving a high degree of overhead expense are reduced or eliminated where possible.

Brief Summary Text - BSTX (13): By means of the present invention, an authorized user at the target computer is able to "download" programs or data, via a telephone line and a programmable remote control module (RCM) connected at each end thereof from a central or host computer. Usage and other accounting data are monitored by the RCM and stored in memory resident therein. At predetermined times, the central or host

computer accesses the RCM for the purpose of "uploading" the usage and other accounting data to the central or host computer.

Detailed Description Text - DETX (49): The key module is encrypted using the Federal Information Processing Data Encryption Standard No. 46, well-known to those of skill in the art, by the data encryption/decryption module 70 of RCM 16. When the rental software is transmitted by the host computer 12 over the telephone network 26, the encrypted key module and the associated OSP module are transmitted as well. Alternatively, the encrypted module, the OSP module and the unencrypted remainder of the rental software may be sent to the customer on floppy disks or magnetic tape by mail or other delivery service. When downloaded from the host computer 12 or loaded from media otherwise provided by a software rental service, the entire rental software package (including the encrypted key module and OSP module) is stored in a peripheral storage device (e.g., hard disk or floppy disk) associated with the target computer 14.

Claims Text - CLTX (47): said host remote control module and said target remote control module including means for automatically downloading computer programs from said host computer to said target computer over said telephone network during off-peak hours, and for uploading elapsed time of use of downloaded computer programs from said target computer to said host computer over said telephone network during off-peak hours; and

Claims Text - CLTX (56): said host remote control module and said target remote control module including means for automatically downloading computer programs from said host computer to said target computer over said communicating means during off-peak hours, and for uploading elapsed time of use of downloaded computer programs from said target computer to said host computer over said communicating means during off-peak hours; and

Claims Text - CLTX (86): 23. In a system having a central means for storage of a plurality of computer programs, and having communicating means for communicating with a computer for downloading said computer programs to said computer and for monitoring the use of said computer programs by said computer, said computer being controlled to transmit at predetermined times data indicating elapsed time of use of downloaded computer programs, a remote control device for controlling and monitoring the use of said downloaded computer programs that include encrypted portions in said computer, said remote control device comprising:

US-PAT-NO: 5734831
DOCUMENT-IDENTIFIER: US 5734831 A
TITLE: System for configuring and remotely administering a unix computer over a network
DATE-ISSUED: March 31, 1998
INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Sanders; James B.	Menlo Park	CA	N/A	N/A

US-CL-CURRENT: 709/223

ABSTRACT: A forms based browser interface system for configuring and administering a network server from a remote location. Using forms, such as hyper-text markup language forms, the system provides a graphical user interface that allows a novice user, unaware of the platform, architecture or even operating system of the network server, to transact administrative tasks on the network server. An interfacing computer, at which the novice user performs administrative tasks upon the network server, is connected to the network server via network connections. The interfacing computer is equipped with a browser program that can display and interact with the forms created by the network server. The forms allow the user to select among various administrative tasks to be performed on the server. The forms also allow the user to input parameters for administration of the server such as new account names when adding new accounts for the server. Once the form input is submitted over the network connections to the network server, scripts within the server pass this information as parameters to appropriate software that complete the execution of the task and may signal to the user at the interfacing computer, through messages on the forms, success or failure thereof.

16 Claims, 8 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 8

--------- KWIC ---------

TITLE - TI (1): System for configuring and remotely administering a unix computer over a network

DATE ISSUED - PD (1): 19980331

Brief Summary Text - BSTX (7): Further, in order to install and configure a UNIX-based computer, the computer must be connected to a monitor and an input device, usually a keyboard with which a UNIX trained technician must manually type commands. Thus, the operating system, system utilities and some software applications that run on the UNIX system cannot be made available for use without the help of a UNIX trained technician working at the site of the server. A UNIX computer with devices such as disk or tape drives requires, for instance, complicated configuration known as "mounting" such that the operating system and software can recognize and use the drives. Similarly, networking interfaces such as a TCP/IP stack or packet

drivers must be configured upon the installation of the operating system and/or the communications software that uses them such that File Transfer Protocol (FTP) or Telnet may operate correctly. When the UNIX computer also operates as a server, then the server software must be correctly installed and configured to run administrative tasks such as file permissions and setting up user accounts. This has typically required a system administrator with experience in such features as shell scripts and daemons (for file systems and E-mail). Even when administrative tasks are made available to a system administrator, however, they must be performed using a terminal at the physical location of the server.

Brief Summary Text - BSTX (8): Thus, there is need for a method and apparatus that circumvents the need for trained technicians in order to configure and remotely administer computers such that persons not trained in the operating system or platform of the computer can perform these tasks graphically and remotely.

Brief Summary Text - BSTX (10): The present invention is a method and system for automating the initial configuration of a computer system and providing for remote ongoing administration of the computer system, particularly when the system is UNIX-based network server.

Detailed Description Text - DETX (15): Among the basic services provided by the packages are Hyper Text Transfer Protocol (HTTP), Dynamic Host Configuration Protocol (DHCP), Internet Mail Protocol (IMAP) and Post Office Protocol (POP). Additionally, the server 10 will have a Domain Name Server (DNS) package providing name-server information such as the IP addresses of a cache server, a primary server and a secondary server. A "sendmail" service is also provided to give Simple Mail Transport Protocol (SMTP) functionality to the users of the server 10. In terms of administrative support, server 10 is configured to provide asynchronous Point-to-Point Protocol (PPP) setup, File Transfer Protocol (FTP) setup, added security (for granting super-user or ordinary user access to the server over the network) and access to newsgroups or Gopher information services, and the ability to add user accounts for mail and web access. In terms of configuration support the following functionality is exemplary: IP addressing, netmask, default route schemes, root passwords, date, time and timezone setting.

Detailed Description Text - DETX (48): HTML forms, such as the "Add User Account" form shown in FIG. 8, allow a user at the remote interfacing computer to input information for adding a new account. Appendix A contains source code showing how this form may be generated. Referring to FIG. 8, input windows, one for the account's login name, another for the account user's real name, another one for the password, and so on are illustrated. Once the information is input into the windows, selecting the menu item "Add", as shown at the bottom of the screen, will "submit" the information over the network 130 and then the appropriate script(s) for that package on the server 10, in this case SUNWuserA, will run and the new user information will be passed as program parameters. The SUNWuserA scripts perform the necessary operations to get the new account ready, such as setting up directories for the new

account, setting access permissions for the user of the new account, setting up a mail spooler or other related tasks. Appendix B contains source code showing how submitted information is processed and how UNIX operations necessary to add a new user account are performed.

Detailed Description Text - DETX (56): Selecting the "To Welcome" menu item takes the user back to the welcome or home page of the server. Each of the other five menu items will transfer the user to a different HTML page, which corresponds to the "home page relevant to preferred task" as mentioned in Step 1300. Selecting the "Manage Users" page allows administrators to add and manipulate user accounts on the server, provide mail service, or create logins for placing web pages. In the case of user accounts, a user account administrative page is then displayed on the screen from which the option "Add User Account" may be selected. Then, as shown in FIG. 8, the new account information can be configured. The "Hosts" page lets administrators add, delete or modify host entries via the domain name server and dynamic host configuration protocol. Selecting the "Advanced Management" menu item provides users with functionality for modifying the server system itself. The "Advanced Management" page has menu items such as:

Detailed Description Paragraph Table - DETL (1): TABLE I

Static Information residing at /opt	
SUNWnak/cgi-bin/	Contains HTML forms work, such as main,.cgi
SUNWnak/html/	Various HTML pages
SUNWnak/icons/	Various xbm icons
SUNWnak/images/	Various GIF/JPEG images
SUNWnak/audio/	Package-related audio files
SUNWnak/bin/audio	Program to play audio files at a specified volume
SUNWnak/bin/config	Program to call configuration programs of other packages
SUNWnak/bin/startup	Run at boot-time to check configuration and update from disk if necessary
vendor	Symbolic link to vendor-specific "personality package"
vendor-proto	Prototype of a vendor package that ships with the computer

US-PAT-NO: 5838907
DOCUMENT-IDENTIFIER: US 5838907 A
TITLE: Configuration manager for network devices and an associated method for providing configuration information thereto
DATE-ISSUED: November 17, 1998
INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Hansen; Peter A.	Houston	TX	N/A	N/A

US-CL-CURRENT: 709/220, 709/221 , 709/223 , 709/224 , 709/225

ABSTRACT: A configuration manager for configuring a network device remotely coupled thereto and an associated computer-implemented method for configuring the network device. The configuration manager includes a configuration script stored in a memory subsystem of a computer system and first and second software modules respectively executable by a processor subsystem of the computer system. The configuration script contains a series of executable instructions for constructing a configuration file and a bootptab file for a first specified type of network device. By executing the instructions contained in the configuration script, the first software module may construct a configuration file suitable for upload to a network device and a bootptab file suitable for identifying the network device. Configuration requests issued by the network device are processed by the second software module by identifying the requesting network device using the constructed bootptab file and configuring the requesting network device by uploading the constructed configuration file thereto.
18 Claims, 17 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 12

---------- KWIC ---------

Abstract Text - ABTX (1): A configuration manager for configuring a network device remotely coupled thereto and an associated computer-implemented method for configuring the network device. The configuration manager includes a configuration script stored in a memory subsystem of a computer system and first and second software modules respectively executable by a processor subsystem of the computer system. The configuration script contains a series of executable instructions for constructing a configuration file and a bootptab file for a first specified type of network device. By executing the instructions contained in the configuration script, the first software module may construct a configuration file suitable for upload to a network device and a bootptab file suitable for identifying the network device. Configuration requests issued by the network device are processed by the second software module by identifying the requesting network device using the constructed bootptab file and configuring the requesting network device by uploading the constructed configuration file thereto.

Application Filing Date - AD (1): 19960220

DATE ISSUED - PD (1): 19981117

Brief Summary Text - BSTX (6): This application generally relates to computer networks and internetworks and, more particularly, to a configuration manager which, from a central location, provides configuration information to remote devices included in a computer network or internetwork.

Brief Summary Text - BSTX (12): Thus, it can be readily seen from the foregoing that it would be desirable to simplify the task of configuring a remotely located network device. It is, therefore, the object of this invention to provide a configuration manager and an associated method of configuring a remote network device from a central location.

Brief Summary Text - BSTX (14): In one embodiment, the present invention is of a configuration manager for configuring a network device remotely coupled thereto. The configuration manager includes a configuration script stored in a memory subsystem of a computer system and first and second software modules respectively executable by a processor subsystem of the computer system. The configuration script contains a series of executable instructions for constructing a configuration file and a bootptab file for a first specified type of network device. By executing the instructions contained in the configuration script, the first software module may construct a configuration file suitable for upload to a network device and a bootptab file suitable for identifying the network device. Configuration requests issued by the network device are processed by the second software module by identifying the requesting network device using the constructed bootptab file and configuring the requesting network device by uploading the constructed configuration file thereto.

Brief Summary Text - BSTX (17): In another embodiment, the present invention is of a computer-implemented method for configuring a remotely located network device. A request for configuration issued by a network device is detected. If a previously constructed configuration file corresponds to the network device issuing the request for configuration, a reply which identifies the configuration file is transmitted to the network device. The configuration file is then transmitted to the network device in response to a request for the identified configuration file. The configuration file is constructed using a configuration script containing a series of executable instructions for constructing a configuration file for a first specified type of network device is provided. The configuration file is then constructed by executing the series of instructions contained in the configuration script. In one aspect thereof, the configuration script includes a first section containing a series of configuration commands. Requests for information are issued by executing the series of configuration commands contained in the first section of the configuration script and information received in response to the requests for information is used to construct the configuration file. The information may also be used to construct a bootptab file which, in addition to the configuration file, contains a unique identifier for the network device.

Drawing Description Text - DRTX (18): FIG. 8 is a flowchart of a method of configuring a remote network device in accordance with another aspect of the present invention; and

Detailed Description Text - DETX (6): For each network device for which a local configuration file has been constructed, the network device configuration tool 10 may also construct a network device configuration file suitable for export to the network device itself. In this manner, remote configuration of network devices is enabled.

Detailed Description Text - DETX (8): The data and programming instruction are stored in the memory subsystem 6 as a series of files which may be selectively accessed by the map editor 14 and/or the configuration guide 18. Files which are accessible to the map editor 14 and/or the configuration guide 18 are configuration scripts 12, map files 16, local configuration files 20 and network configuration files 22. The configuration scripts 12 identify the types of network devices and network entities which may be placed on the network configuration map and interconnected with other network entities and network devices. The configuration scripts 12 also identify the network devices which are configurable by the network device configuration tool 10 and contain information necessary to construct configuration files for those network devices. If a particular network device does not have a configuration script, a configuration file cannot be constructed by the network device configuration tool 10. The map files 16 contain a series of network configuration maps, each comprised of a series of interconnected network devices and network entities, constructed using the network device configuration tool 10. The local configuration files 20 contain information which, if uploaded to the corresponding network device 26, would enable configuration of that device. If local configuration files 20 are constructed for the network devices illustrated on the network configuration map(s) 16 produced using the network device configuration tool 10, such local configuration files 20 are associated with the corresponding network device such that they may be directly accessed from the network configuration maps 16.

Detailed Description Text - DETX (10): It is contemplated that the network device configuration tool 10 would be installed in the computer system 2 operated by a network administrator and that plural network devices 26 and other network entities, only one of which is shown in FIG. 1B for ease of illustration, would be coupled to the network device configuration tool 10. Utilizing the network device configuration tool 10, the network administrator may build a representative network configuration map for the network. The network administrator may then configure remotely located network devices by uploading configuration files constructed during the process of building the network configuration map to the devices. Thus, by using the network configuration tool, the network administrator can, from a central location, design a suitable configuration network and then configure any number of remotely located devices included in the network.

Detailed Description Text - DETX (21): If a request to configure a device placed on the network configuration map is received, the map editor 14 transfers the name and

connection information for the device to the configuration guide 18 and instructs the configuration guide 18 to perform the requested configuration task. For example, if configuration of a network device is requested, the configuration guide 18 will retrieve the configuration script 12-N for that type of network device and execute the instructions contained in the guided configuration section 40 thereof. Using the information provided by the configuration script 12, the map editor 14 and input provided by the network administrator in response to execution of the instructions contained in the guided configuration section 40, the configuration guide 18 builds a local configuration file, associated with the device, for use by the network administrator and a corresponding network configuration file suitable for upload to the network device to enable configuration of the network device.

Detailed Description Text - DETX (26): The edit device command accesses the configuration information associated with a selected network device. The view/configure command displays a view of the backplane of a selected configured network device or, if the selected network device is unconfigured, defaults to the configuration dialog set forth in greater detail below. The delete device command removes a selected network device or entity from the network workspace. The all ports configured, update configuration provides access to a selected device's configuration file. The retrieve configuration file allows the network administrator to directly access a configuration file stored in the memory subsystem 6 while the associate configuration command permits the network administrator to append a configuration file to a device. The telnet to the device command initiates an in-band transfer of configuration information from the network device configuration tool 10 to the network device 26.

Detailed Description Text - DETX (46): While constructing a local configuration file for a device, the network device configuration tool 10 also constructs a bootptab file for the device. The bootptab file is particularly useful in those situations where the network administrator decides not to upload the configuration file upon completing the construction thereof, for example, if the network device is unconnected, powered down or otherwise unavailable. A bootptab file for a device contains the serial number for the device to be configured, an IP address to assign to the device to be configured and the configuration file to be uploaded to the device. As will be more fully described with respect to FIGS. 8-9, below, the bootptab file provides information necessary for unattended remote configuration of network devices as they are connected to the network.

Claims Text - CLTX (1): 1. For a computer system having a processor subsystem and a memory subsystem coupled by a system bus for bi-directional exchanges therebetween, a configuration manager for configuring a network device remotely coupled thereto, said configuration manager comprising:

Claims Text - CLTX (5): 2. A configuration manager for configuring a network device remotely coupled thereto according to claim 1 wherein information received by said first software module in response to said requests for information is used to construct said configuration file and said bootptab file.

Claims Text - CLTX (6): 3. A configuration manager for configuring a network device remotely coupled thereto according to claim 1 and wherein said second section of said configuration script further comprises:

Claims Text - CLTX (9): 4. A configuration manager for configuring a network device remotely coupled thereto according to claim 3 and wherein said first section of said configuration script further comprises:

Claims Text - CLTX (12): 5. A computer-implemented method for configuring a remotely located network device, comprising the steps of:

Claims Text - CLTX (19): 6. A computer-implemented method for configuring a remotely located network device according to claim 5 and further comprising the steps of:

Claims Text - CLTX (22): 7. A computer-implemented method for configuring a remotely located network device according to claim 6 and further comprising the steps of:

Claims Text - CLTX (24): 8. A computer-implemented method for configuring a remotely located network device according to claim 7 wherein the step of determining if said configuration file corresponds to said network device issuing said request for configuration further comprises the steps of:

Claims Text - CLTX (27): 9. A computer-implemented method for configuring a remotely located network device according to claim 5 wherein the step of providing a configuration script containing a series of executable instructions further comprises the step of:

Claims Text - CLTX (30): 10. A computer-implemented method for configuring a remotely located network device according to claim 9 and further comprising the step of:

Claims Text - CLTX (33): a plurality of configuration scripts for various different types of network devices, said configuration scripts being stored in said memory subsystem, and each of the configuration scripts being used to construct a configuration file and a bootptab file, the bootptab files being constructed are used to assist with remote configuration of network devices including at least one connection statement for each connection port of the associated one of the network devices, the connection statements included in said configuration scripts comprise connection rules that specify permissible connections between the ports of the various different types of network devices and other of the network devices, the connection rules include an identifier for the associated port and a list of network devices that are permitted to connect to the associated port;

Claims Text - CLTX (41): a plurality of configuration scripts for various different types of network devices, said configuration scripts being stored in said memory subsystem, and each of the configuration scripts being used to construct a configuration file and a bootptab file, the bootptab files being constructed are used to assist with remote configuration of network devices including at least one connection statement for each connection port of the associated one of the network devices;

US Reference Patentee Name - URNM (6): Sanchez-Frank et al.

US Reference Group - URGP (6): 5394522 19950200 Sanchez-Frank et al. 395/159

US-PAT-NO: 6012100
DOCUMENT-IDENTIFIER: US 6012100 A
TITLE: System and method of configuring a remotely managed secure network interface
DATE-ISSUED: January 4, 2000
INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE
Frailong; Jean-Marc	Palo Alto	CA	N/A
McManis; Charles	Sunnyvale	CA	N/A
Price; Charles A.	San Jose	CA	N/A
Herbert; Mark James	San Jose	CA	N/A
Gastinel; Jean Antoine	Mountain View	CA	N/A
Tardo; Joseph John	Palo Alto	CA	N/A

US-CL-CURRENT: 709/250, 709/220 , 709/223

ABSTRACT: The present invention discloses a network interface device for connecting a client computer system to an external network. The network interface device is configured for the client system by automated procedures and protocols initiated from a remote server. Software programs within the network interface device provide transparent communication between the client computer system and services available on the external network. Similar software programs and a configuration database within the network interface device provide transparent communication between the client computer system and the remote server.
21 Claims, 15 Drawing figures
Exemplary Claim Number: 15
Number of Drawing Sheets: 15

---------- KWIC ---------

Application Filing Date - AD (1): 19970714

DATE ISSUED - PD (1): 20000104

Brief Summary Text - BSTX (10): According to one aspect of the present invention, a network interface device is provided to connect a client computer network to an external network. The network interface device is provided to the client user in an initially unconfigured state. The network interface device is configured for the client system by automated procedures and protocols initiated from a remote server. The remote server provides and maintains the client information in a secure database. The use of a secure database and automated procedures minimizes the amount of input required from the user. The network interface device contains application program interfaces which facilitate communication between the client computer system and services available on the external network. The network interface device also contains a configuration database which stores data and parameters related to the configuration of the network interface device. Through the use of the configuration database and the

resident application program interfaces, the remote server is able to automatically upgrade or reconfigure the network interface device without user intervention.

Detailed Description Text - DETX (7): In one embodiment of the present invention, the various physical network interface devices, security functions, and service interfaces are replaced by a single integrated network interface device, hereinafter referred to as a `gateway interface device`. This integrated gateway interface device provides a single point of connectivity for various different types of data communication lines, such as Ethernet and ISDN, and contains a configuration database for the storage of parameters associated with the operation of the network interface. The gateway interface device also contains application program interfaces (API's) for transparent communication between the client LAN and various internet services. The gateway interface device further provides connectivity to a remote server process which provides remote initialization, configuration, and upgrades of the gateway interface device without necessitating extensive user interaction.

Detailed Description Text - DETX (9): The remote server 206 represents central facility for providing convenient and efficient configuration and maintenance of the gateway interface device. In one embodiment of the present invention, the remote server 206 (hereinafter referred to as the "remote management server") is connected to ISP 204 and maintains a dynamic dialog with ISP 204 to configure and maintain gateway interface device 208 in client network 220. Remote management server 206 interacts with gateway interface device 208 to provide configuration information and upgrade parameters required by the gateway interface device 208. In this manner, remote management server 206 basically serves as a repository for information required by the gateway interface device 208. Such information may include configuration information related to LAN 210, internet address blocks, internet domain names, and data related to the physical and logical interfaces between the client network 220 and ISP 204.

Detailed Description Text - DETX (11): Remote management server 206 and gateway interface device 208 contain security information such as passwords and encryption keys that are used to establish a trust relation sufficient to ensure secure remote configuration and upgrade of gateway interface device 208. By providing a configuration management function within remote management server 206 which is registered with an ISP 204, it is possible to download configuration and upgrade information and parameters to gateway interface device 208 at the time the gateway interface is first installed between the client network 220 and the telephone client 204. This eliminates the requirement that the network administrator program the network interface device with such configuration and initialization information. This system thus greatly reduces the amount of work required to connect client network 220 to an internet.

Detailed Description Text - DETX (20): The third layer of system software 400 is the run-time section 406. Runtime section 406 contains the management daemons and services required for system control. In one embodiment of the present invention, run-

time section 406 is implemented as a console-less version of a standard operating system. The implementation of a console-less operating system runtime allows the system software to operate without user intervention, thus facilitating the remote access capabilities of the present invention. This system also provides an interface to existing network services which are wrapped in a management layer to allow them to be plugged in or interfaced to the system without requiring user intervention or configuration. Such services that may interface with the system software include web service, electronic mail service, and other similar computer programs and application programs.

Detailed Description Text - DETX (32): Several different service managers may be available. A minimal set of service managers for a typical internet access scenario may include a domain name service (DNS) manager, HTTP manager, electronic mail manager, IP manager, ISDN manager, and system manager, among others. The implementation of service managers allows the use of unmodified services. The service managers provide a consistent interface and minimize the necessary changes to a service to integrate the service in the system.

Detailed Description Text - DETX (41): The combination of a security framework, configuration manager API, service managers and diagnostic reporting capabilities within the runtime layer 406 of system software 400 creates a generic framework for interfacing with various network services through a single user interface. It also allows remote management of the gateway interface device, and provides an efficient mechanism for initially configuring, upgrading, or reconfiguring the gateway interface device.

Detailed Description Text - DETX (44): The initialization protocol utilized by the gateway interface device provides a method by which the gateway interface device and the gateway computer of the client LAN are configured for internet access from a remote server with minimal user interaction. In one embodiment of the present invention, the remote server (remote management server) communicates with the gateway interface device through the external network medium, but is viewed as a virtual device in terms of configuration and remote management from the point of view of the client network. The initialization protocol is used when the client network orders internet access from an Internet Service Provider and receives the gateway interface device.

Detailed Description Text - DETX (50): The gateway interface device then initiates a remote procedure call (RPC) communication session with the gateway interface device and provides an encryption key. This RPC communication is established by the gateway interface device to request configuration information from the remote management server. As part of this communication session, the remote management server provides a configuration file to the gateway interface device, step 928. In one implementation, the configuration file may be in the form of a script which is executed locally in the gateway interface device, step 932. This step configures the gateway interface device by loading specific parameter values in the appropriate locations of the data store. The

gateway interface device writes configuration values into the configuration manager, step 934. Upon completion of the configuration process, the gateway interface device transmits a message to the remote management server verifying successful configuration, step 936. To conclude the initialization process, the remote management server confirms the gateway interface device verification and marks the registration key as used, step 938. This step prevents unauthorized re-use of the registration key.

Detailed Description Text - DETX (59): During the time period specified by the fetch time window, the gateway interface device retrieves the upgrade package from the specified FTP site, step 1016. Upon retrieving the upgrade package, the gateway interface device executes the pre-install script to verify the possibility and appropriateness of the upgrade, step 1018. The pre-install script determines whether it is physically possible to upgrade the software within the gateway interface device. The gateway interface device can reject an upgrade on the basis of factors such as insufficient memory to perform the upgrade, or an attempted upgrade to a software version which is already present on the gateway interface device. The pre-install script ensures that an upgrade operation either completely fails or completely succeeds so that a gateway interface device or a client network is either fully upgraded, or left in the original state with regard to the version of the gateway software.

Detailed Description Text - DETX (62): If, however, in step 1022 the gateway interface device determines that the upgrade and reboot were successful, the gateway interface device then executes the post-install script and notifies the remote management server of the upgraded status, step 1030. The remote management server stores this upgraded status as part of the configuration information related to that particular gateway interface device. The post-install script contains commands for resolving references within the upgraded software, as well as recording the upgraded version number in appropriate places for the configuration manager.

Detailed Description Text - DETX (68): The reconfiguration protocol between the remote management server and the gateway interface device is used when the gateway interface device is to be reconfigured in some manner. Unlike an upgrade which is the substitution of all of the software components within the gateway interface device, reconfiguration involves only an upgrade or changes to parameters in the data store of the gateway interface device. The reconfiguration package made available to a gateway interface device includes only an apply time window. No fetch time window or encryption key is required.

Detailed Description Text - DETX (70): FIG. 12 is a flow chart which illustrates the reconfiguration process. The reconfiguration process starts with the remote management server sending reconfiguration notification messages to eligible target gateway interface devices, step 1202. In one embodiment of the present invention, the reconfiguration package simply consists of data store operations for new parameters, as well as an apply time window. Upon receiving the reconfiguration notification message, the gateway interface device verifies the reconfiguration request. If the reconfiguration request is not acceptable, the gateway interface device notifies the remote management

server. If, however, the reconfiguration request is valid, the gateway interface device records the notification message, step 1204. In step 1206, the gateway interface device writes the new parameters specified in the reconfiguration message to the data store at the time specified by the apply time window. In step 1208 the gateway interface device verifies that the reconfiguration was successful. If the reconfiguration is not successful, the gateway interface device notifies the remote management server of a reconfiguration problem, step 1210, and then automatically rolls back to the state prior to the reconfiguration request, step 1212. If, however, in step 1208 it was determined that the reconfiguration was successful, the gateway interface device notifies the remote management server of its reconfigured status, step 1214.

Detailed Description Text - DETX (91): Thus, a method and apparatus have been described for allowing the remote initialization, configuration and upgrade of a network interface device. Although the present invention has been described with reference to specific exemplary embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the invention as set forth in the claims. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.

US-PAT-NO: 6067582
DOCUMENT-IDENTIFIER: US 6067582 A
TITLE: System for installing information related to a software application to a remote computer over a network
DATE-ISSUED: May 23, 2000
INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Smith; Benjamin Hewitt	Belmont	MA	N/A	N/A
Smith; Fred Hewitt	Belmont	MA	N/A	N/A

US-CL-CURRENT: 710/5, 709/203 , 717/177

ABSTRACT: A system and method is disclosed for distributing, registering and purchasing software application and other digital information over a network. Each software application is embedded with an agent module which communicates with a remote server module in a server attached to the network. The server module interacts with the user that is requesting installation of the software application and upon verification of billing or other constraints, the server module enables the agent module to proceed with installation. Subsequent to installation, the agent module monitors the server module and informs the user if an update to the software application is available.
24 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 5

---------- KWIC ---------

Abstract Text - ABTX (1): A system and method is disclosed for distributing, registering and purchasing software application and other digital information over a network. Each software application is embedded with an agent module which communicates with a remote server module in a server attached to the network. The server module interacts with the user that is requesting installation of the software application and upon verification of billing or other constraints, the server module enables the agent module to proceed with installation. Subsequent to installation, the agent module monitors the server module and informs the user if an update to the software application is available.

Application Filing Date - AD (1): 19960813

TITLE - TI (1): System for installing information related to a software application to a remote computer over a network

DATE ISSUED - PD (1): 20000523

Brief Summary Text - BSTX (3): Digitally encoded information, or software, is one of the most economically important commodities of the era. The ease and economy with which perfect copies can be made and distributed has promoted the spread of software and related technologies through traditional commercial channels such as retail and

mail-order sales. More recently, non-traditional distribution channels such as distribution over networks of interconnected computers such as the Internet have become more viable. These non-traditional distribution channels have made it difficult for software creators and copyright holders to regulate the use of their creations or to receive payment and registration information from their users. Consequently, software producers forfeit substantial revenues and valuable information about their customer base and potential markets while businesses and universities find themselves subject to legal prosecution and intimidation for software piracy.

Brief Summary Text - BSTX (15): The aforementioned and other objects of the invention are achieved by the invention which is a system for installing a software application to a remote computer via a network. The network is one which has at least one content server located thereon, which serves data to a plurality of attached computer clients. This network model is intended to include both intranets and internets. That is, the network may be an internal corporate network, an intranet, or a global network of networks such as the Internet, for example. The system comprises a server module and an agent module.

Brief Summary Text - BSTX (17): The agent module can be embedded in more than one type of software application, and is actuatable by the remote computer to initiate installation of the software application on the remote computer. Upon initiation, the agent module electrically communicates with the server module which selectively enables the installation. In the case of a commercial distribution of the software over the Internet, for example, the user, upon finding a software application which the user wishes to purchase, the user selects the software application. An agent module would then communicate with the server module.

Brief Summary Text - BSTX (20): The user of the remote computer then accepts the pricing, confirms acceptance of license terms and inputs user information, all of which is then transferred back to the server module as identification information. Upon verified receipt of the user's information, the server module then enables the installation of the software application by communicating with the agent module, either the agent module embedded in the software application or the plug-in embedded in the browser. Installation of the software application can then proceed over the network.

Detailed Description Text - DETX (3): Referring now to FIG. 1, a remote computer 10 is shown having an electrical connection 12 to a network 14. The remote computer 10 can be a personal computer, such as an IBM compatible or a MAC, can be a work station, or any other such computer that is adapted to communicate over a network 14. The electrical connection 12 is used generically to indicate a physical connection to a client/server network. Though such a connection can take any of various forms and use any of numerous protocols, in the preferred embodiment communication via the electrical connection 12 uses Transfer Control Protocol/Internet Protocol ("TCP/IP"). TCP/IP is preferred as it is the communication protocol suite required to communicate over the Internet. Communication over the Internet is desirable because the Internet is a global interconnection of multiple content servers which are freely communicable to

each other and accessible by an unlimited group of remote computers. For illustration purposes, the network 14 will be assumed to be the Internet, though other possibilities exist, such as electronic mail networks utilizing X.25 protocols.

Detailed Description Text - DETX (7): In another embodiment, the agent module 22 embedded in the software application 20 is inactive until after the installation of the software application 20 on the remote computer 10. In this embodiment, a second agent module, a plug-in module 23, is installed on the remote computer 10 and used to access the virtual store 18. The plug-in module 23 is made available on the Internet or other well known resources or by other well known methods, for installation on the remote computer 10 by the user. The plug-in module 23 is preferably disposed in a software package 19, a browser, which the user of the remote computer 10 uses to access the virtual store 18. When the agent module 22 is actuated the agent module 22 and the plug-in module 23 have the same functionality, and the agent module 22 and plug-in module 23 are used interchangeably by this invention, although for clarity the embodiments will refer to one or the other module.

Detailed Description Text - DETX (12): In the preferred embodiment, the dialog box displays information prepared by an independent auditor relative to assuring that the installation software performs only certain limited functions strictly necessary for the software installation and does not examine or transfer other data from the remote computer 10. The dialog box also gives the user the option of verifying the credentials of the installer.

Detailed Description Text - DETX (16): Once the billing information is verified and the user information is recorded in the database maintained by the server module 26, the server module 26 transmits an enabling command to the plug-in module 23 which allows transmission of the software application 20 to the remote computer 10. The software application 20 is transmitted as an installation program which is then installed locally to the remote computer 10.

Detailed Description Text - DETX (20): The user is then queried whether installer verification is required 38. If so, then a code given to the user by the plug-in module 23 is input 40. The code helps determine information appropriate to the software application 20 and the installer, which should be transmitted to the user. If the installer information is sufficient and the installer is verified 41, then the installation process is continued. Otherwise, the installation is terminated 64.

Detailed Description Text - DETX (22): As previously described, in the preferred embodiment the user is provided with information by the independent auditor regarding the limited functionality of the installing program. The user is given the option of verifying the credentials of the installer. If the user chooses this option, the user is given the option of connecting to the auditor via a Web Browser such as NETSCAPE or directly by the plug-in module 23. Once connected to the audit module housing the verification program 34 disposed on the audit server 30 provided by the independent auditor 36, the user will have the opportunity to review the assurances provided by the

auditor and to verify that the installer is known to the auditor. To perform the verification, the user would enter a code which might be a checksum for the installer and the software application 20 of the installer, would appear on the user screen 10. Using this code, the auditor would verify that the installer is certified by the auditor. At this stage, the user also has the option of registering with the installer by entering user information such as name, address and phone number, regardless of whether the user will purchase a software application from the installer. After the verification, the user would also have the option of terminating the installation.

Detailed Description Text - DETX (25): The billing information is then transferred back to the server module 26, which verifies the billing information 46. Such verification in the preferred embodiment is done by communicating the numbers to a central source of verification in much the same manner as is done for conventional transactions. That is, the credit card number is transmitted to a credit card number verification service and a verification code is transmitted back. If the billing information is not accepted 48, then the user is invited to input new billing information 44. If the billing information is accepted 50, then the plug-in module 23 is sent an enabling signal which allows transfer of installation modules 52 of the software application 20 to the remote computer 10. The installation modules are generally executable modules which are created by the server at the time of a request by the plug-in module 23 so as to contain only the particular product options which the user has purchased. Therefore, the executable code can be configured so that it will only operate on the remote computer 10 for which the user has purchased the application software 20.

Detailed Description Text - DETX (26): The executable code is transmitted as a self-extracting executable as is well known in the art. The plug-in module 23 then executes the self-extracting executable which proceeds to automatically install the software application 20 on the remote computer 10.

Detailed Description Text - DETX (27): The user then follows the procedure proscribed therein to install the software on the remote computer 10. The server module 26 during this process monitors the installation to verify the installation 54. Upon completion, the plug-in module 23 in the software application transmits installation information back to the server module 26. If the installation failed, or was unsuccessful 56, then the installation logs and the identification information are transmitted to a technical department 58 of the installer or the developer such that contact can be made to the user directly. The transmission to the technical department can be by any known communication method including manual contact. In the preferred embodiment, however the technical department would be disposed upon the network and in electrical communication with the server module 26. The technical support person would then have, prior to making any contact with the user, complete information related to the hardware and software and the installation attempt, all prior to contacting the user thus expediting the support process.

Detailed Description Text - DETX (34): A check is performed to confirm that the update is being requested by the same remote computer 10 on which the software application 20 was originally installed 80. Then, the server module 26 checks for the availability of a newer or a full version 78. In this way, piracy is inhibited in that the same remote computer 10 must be requesting the update as was the one that originally requested the software application 20. Under some circumstances, the remote computer 10 may change for reasons other than pirating software. Such circumstances can include replacing the computer with a more modern computer or transferring the software application 20 pursuant to the terms of the license agreement to a third party. Under these circumstances, the user may transfer the information specific to the remote computer 10 to the new computer as long as verification is made that the old computer either no longer exists or is no longer loaded with the software application 20.

Detailed Description Text - DETX (36): If this request is not made and it is determined that the user is pirating the software 82, then a signal is sent from the server module 26 to the agent module 22 in the remote computer 10 to disable the program 94. The program will then no longer be usable by the remote computer 10 and only the complete new installation including a purchase of the software will re-enable the software.

Detailed Description Text - DETX (38): If the software update is more than simply a maintenance update, there may be additional billing necessary. The user is then enabled to use the previous billing information or input new billing information 88. Upon verification of the billing information, the new version is then installed 90 and the procedure is complete 92.

Detailed Description Text - DETX (42): The system administrator for the corporate network 102 would generally purchase a predetermined number of licenses for the software application 110 using a method similar to that previously described or upload them directly from a vendor's disk. The user would then access the software application 110 using the client computer 104 via the corporate network 102. Requesting installation of the applications software 110 causes the agent contained therein to contact a server module 114 which, in this embodiment, is stored on the hard disk 108 of the corporate server 100. The server module 114 monitors the number of licenses and, if more are available, then enables the installation.

Detailed Description Text - DETX (56): The auditor will verify that the TCP/IP send function is encapsulated in a class XAP.sub.-- Ship, and appears only inside this class. The auditor will verify that a XAP.sub.-- Ship object can only be created from, a XAP.sub.-- Packet, and that therefore, without a XAP.sub.-- Packet object the application will cannot ship data over TCP/IP. The auditor will verify that the TCP/IP ship function only ships the data provided to it by XAP.sub.-- Packet. From these observations, the auditor will be able to conclude that the application can only ship data that can be extracted from a XAP.sub.-- Packet.

Detailed Description Text - DETX (57): From examination of the sources, the auditor will observe that a. XAP.sub.-- Packet can only be composed of XAP.sub.-- Record objects. C++ has facilities for defining insertions and extractions from classes, which can limit the insertions and extractions to certain other classes. The XAP.sub.-- Record classes will be defined in the application. The auditor will examine each of these classes. From examination of the individual XAP.sub.-- Record classes the auditor will conclude that these classes can be composed only of certain XAP.sub.-- Data objects. Again the XAP.sub.-- Record classes will have defined insertions and extractions which will limit the data that can be put into these records to XAP.sub.-- Data objects. Finally, the auditor will examine the various defined, XAP.sub.-- Data objects, and will observe that the XAP.sub.-- Data objects represent the data needed to perform an installation but will not permit other data that might violate a user's privacy. For example, the data objects could have predefined limits on overall length and predefined data values which would render impossible the copying of correspondence from the user's personal computer. The data objects would be limited as to size and content so that bitmaps or spread sheets, for example, could not be copied from the user's personal computer.

Claims Text - CLTX (1): 1. A system for installing information related to a software application to a remote computer via a network having at least one content server, comprising:

Claims Text - CLTX (3): an embedded agent module embedded in a version of the software application, the embedded agent module, when the version of the software application is installed on the remote computer, running under the control of the version of the software application to communicate with the server module;

Claims Text - CLTX (5): means for installing the version of the software application with the embedded agent module on the remote computer;

Claims Text - CLTX (21): an audit module disposed on the audit server, the audit module in electrical communication with the server module for communicating to the remote computer information regarding the functionality of the installation file.

Claims Text - CLTX (31): 17. A method for installing information related to a software application on a remote computer via a network, the method comprising the steps of:

Claims Text - CLTX (33): installing the software application with the embedded agent module on the remote computer;

Claims Text - CLTX (50): 24. A method for auditing an agent module to verify the installation of a software application on a remote computer by the agent module comprising the steps of:

Claims Text - CLTX (58): verifying the predetermined objects are necessary to the functionality of the installation of the software application.

US-PAT-NO: 6108420
DOCUMENT-IDENTIFIER: US 6108420 A
TITLE: Method and system for networked installation of uniquely customized, authenticable, and traceable software application
DATE-ISSUED: August 22, 2000
INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Larose; Gordon Edward	Ottawa	N/A	N/A	CA
Allan; David Ian	Ottawa	N/A	N/A	CA

US-CL-CURRENT: 705/59, 380/30

ABSTRACT: A method to create, distribute and install on an installation computer a uniquely customized instance of a software application that is authenticable and traceable to a particular user. A secure distribution agent resident on a distribution computer collects identifying information, and calculates a cryptographic signature of the software application and identifying information. The identifying information and cryptographic signature are embedded in the software application by the secure distribution agent. The software application with embedded data is transmitted via a distribution channel to the installation computer. A user installation agent resident on the installation computer manages the installation of the software application with embedded data on the installation computer. Prior to installation, the user installation agent may use the cryptographic signature to verify that the software application, and the identifying information are authentic and have not been tampered with.
17 Claims, 10 Drawing figures
Exemplary Claim Number: 15
Number of Drawing Sheets: 10

---------- KWIC ---------

Abstract Text - ABTX (1): A method to create, distribute and install on an installation computer a uniquely customized instance of a software application that is authenticable and traceable to a particular user. A secure distribution agent resident on a distribution computer collects identifying information, and calculates a cryptographic signature of the software application and identifying information. The identifying information and cryptographic signature are embedded in the software application by the secure distribution agent. The software application with embedded data is transmitted via a distribution channel to the installation computer. A user installation agent resident on the installation computer manages the installation of the software application with embedded data on the installation computer. Prior to installation, the user installation agent may use the cryptographic signature to verify that the software application, and the identifying information are authentic and have not been tampered with.

Application Filing Date - AD (1): 19970410

DATE ISSUED - PD (1): 20000822

Brief Summary Text - BSTX (4): With the increasing importance and reliance on networked computer environments such as the Internet, Electronic Software Distribution (ESD) is assuming an increased importance as a means of distributing software applications to users. The on-line infrastructures currently in place enable users to purchase and install software applications without the need for physical delivery of shrink-wrapped software. Typically, a software publisher will prepare a master of the software application for electronic distribution. A customer will then go on-line and submit an order to purchase the software application, which will be received and fulfilled by the publisher. The customer will then download the software application and install it to his/her own computer.

Brief Summary Text - BSTX (16): The method and system disclosed herein provides for a user installation agent (UIA) resident on an installation computer to establish a connection through a distribution channel to a secure distribution agent (SDA) resident on a distribution computer. The UIA and/or SDA prompt the user to input identifying information that, together with business related information such as licensing terms, etc., is used to create a unique data set that is embedded in the desired software application by the SDA. By the use of a cryptographic hash algorithm, and private/public key cryptography wherein a private key is only known to the SDA, a cryptographic signature of the desired software application and embedded data set is calculated and also embedded into the software application. The software application with embedded data and cryptographic signature is transmitted via a distribution channel to the installation computer where it is installed on the installation computer. Optionally, the installation computer may use the cryptographic signature to verify that neither the software application, nor the embedded data have been tampered with. Public key(s) used to decrypt the cryptographic signatures may be transmitted to the installation computer with the software application, or by any other means, such as e-mail, Internet bulletin boards, etc. Following installation, the embedded data and cryptographic signature are used in a variety of ways, such as to provide a means to trace the software application to the user, to police the continued integrity of the software application, to ensure that license conditions continue to be met, to perform virus checking, or automatic upgrading of the software application itself.

Drawing Description Text - DRTX (8): FIG. 5 is a block diagram showing the means of extracting and authenticating embedded data from an installed distribution file;

Detailed Description Text - DETX (11): Though the description of the present invention implies that the "user" is an individual user of the software application 15 to be installed on a personal computer, persons skilled in the art will appreciate that the present invention would also operate in the context of a networked end-user environment, where the "user" was a network administrator responsible for installing software on a central server for use by a number of end users.

Detailed Description Text - DETX (29): The method illustrated in FIG. 3A discloses a one-step process wherein cryptographic signature 174 is ascertained for the original distribution file 130 and the embedded data 140. An optional method, such as that

illustrated in FIG. 3B, would be to employ a two-step process wherein a cryptographic signature 172 of the embedded data 171 is first produced using the same algorithm described in step 2 above. This embedded data cryptographic signature 172 is then itself embedded into the original distribution file 130. The original distribution file 130, embedded data 171, and embedded data cryptographic signature 172 are then input to the second cryptographic step, wherein an overall cryptographic signature 176 is ascertained using the same algorithm described in step 2 above. The benefit of the two step process is that it augments the capabilities of the system and method of the present invention to authenticate and detect tampering in the software application installed on the installation computer. For example, separate cryptographic public/private key pairs could be provided for the two cryptographic signatures 172, 176. Furthermore, the two-step process allows the embedded data 171 to be extracted and authenticated, even if the original file contents 173a, 173b have been corrupted.

Detailed Description Text - DETX (52): FIG. 5 illustrates the means of authenticating and extracting user data from an installed aggregate distribution file 15 to verify that neither the original file contents 173a, 173b, nor the embedded data 171 have been tampered with. This step is optional to the operation of the present invention because the installed aggregate distribution file 15 may be run by the user without authentication. It should be understood that the authentication procedures described below can be done either before or after the installation is completed. If authentication is done prior to installation on the installation computer, then the following procedures are directed by the UIA 203 to the aggregate distribution file 170, instead of the installed aggregate distribution file 15.

Detailed Description Text - DETX (53): The process illustrated in FIG. 5 is in relation to an installed aggregate distribution file 15 constructed using the two-step process illustrated in FIG. 3B. The principles of authenticating and extracting user data from an installed aggregate distribution file 15 constructed using the one-step cryptographic process illustrated in FIG. 3A, or the variant of the two-step process illustrated in FIG. 3C, are the same as those described below, with appropriate modifications, depending on the nature of the cryptographic signatures to be compared.

Detailed Description Text - DETX (68): FIG. 6 is a flow-chart of a summary of the procedures described in relation to FIGS. 2, 3A, 3B, 3C, 4 and 5. It should be noted that the public key 152 used to authenticate the integrity of the installed aggregate distribution file 15 could be delivered to the UIA 200 by any means since it is not a secret and might be useful for more than one purpose. For example, the public key may be embedded in the aggregate distribution file 170, it may be explicitly sent to the user as a separate file or message, or it may be obtained automatically by the installation computer from a network trusted authority (e.g. Verisign.TM. Inc.)

Detailed Description Text - DETX (70): Although the present invention has been described with reference to the preferred embodiments, one of ordinary skill in the art will recognize that a number of variations, alterations and modifications are possible. In FIG. 8 there is an illustration showing the various uses of the installed aggregate

distribution file 15. After installation and authentication by the UIA 200, the installed aggregate distribution file 15 may run normally without making use of the embedded data 171 in any way. To ensure licence compliance, the installed aggregate distribution file 15 may also be run in association with a license-enforcement program that verifies that any license terms comprising part of the embedded data 171 are being complied with. The embedded data 171 and cryptographic signatures 172, 174, 175, 176 (depending on the manner in which the aggregate distribution file 170 was constructed) may also be used as an input to a virus checker that may perform an integrity check on the installed aggregate distribution file 15 by using the public key 152 and the same known cryptographic signature algorithm as was employed by the SDA 100. Each time the installed aggregate distribution file 15 is run, the authentication and reading program 400 shown in FIG. 5 may also be run, either by itself, or in association with an authenticating loader that would reject tampered files, and would not permit a tampered installed aggregate distribution file 15 to be run. The embedded data 171 may also be used simply for display to the user.

Detailed Description Text - DETX (71): The method and system disclosed herein can also be used to upgrade an installed aggregate distribution file 15 present on an installation computer. In this case, the UIA 200 and the SDA 100 would verify of the license status of the installed aggregate distribution file 15 present on the installation computer, and then invoke the method and system disclosed herein to construct, deliver and install an upgraded version of the installed aggregate distribution file 15 to the installation computer. The capability to invoke the upgrading feature of the present invention could be done at the request of the user, or it could be invoked automatically upon detection by the UIA 200 of the availability of a new version of the original distribution file 130.

Detailed Description Text - DETX (72): The uniqueness of the installed aggregate distribution file 15 can be used to restrict its operation to a specific central processing unit (CPU) on the installation computer. The identification of the CPU for these purposes would be done by the UIA 200 during the stage of gathering data 32, 34 for transmission to the SDA 100.

Claims Text - CLTX (21): 11. The method of claim 9, wherein prior to the step of installing the software application, the installation computer authenticates the integrity of the software application.

Claims Text - CLTX (22): 12. The method of claim 11, wherein the installation computer uses the cryptographic signature to authenticate the integrity of the software application.

US-PAT-NO: 6195694
DOCUMENT-IDENTIFIER: US 6195694 B1
TITLE: Server for reconfiguring control of a subset of devices on one or more kiosks
DATE-ISSUED: February 27, 2001
INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Chen; Shuang	Somers	NY	N/A	N/A
Fujisaki; Tetsunosuke	Armonk	NY	N/A	N/A
Kobayashi; Makoto	Machida	N/A	N/A	JP
Ohshima; Mitsuru	Yamato	N/A	N/A	JP
Yoshida; Yoichi	Machida	N/A	N/A	JP

US-CL-CURRENT: 709/220, 709/203 , 709/204 , 709/219 , 709/221

ABSTRACT: A server system that is connected to one or more networks, e.g., the Internet, corporate or government intranets, extranets, etc. The server has one or more application files or configuration sets that the server serves to from one or more kiosks on the network. The configuration sets are application specific. (An application is a use for which the kiosks are configured or reconfigured.) One or more of the files in the configuration sets include one or more embedded (control) programs that are used to control the local APIs of one or more of the devices on the kiosk. In this way, the devices are controlled to configure the kiosk to perform the application.
23 Claims, 18 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 14

---------- KWIC ---------

Application Filing Date - AD (1): 19971119

DATE ISSUED - PD (1): 20010227

Brief Summary Text - BSTX (10): The prior art also has combined kiosks with the internet. This kiosk has a browser which displays the HTML pages on the screen of the kiosk. The screen displayed on the kiosk is controlled by the hyperlinks selected by the user. These kiosks are suitable for information access where the client/user can browse through the information provided by selecting "soft" buttons that invoke a hyperlink. These kiosks can also be used for certain personal communications like e-mail. In these systems, the screens are usually specially designed to present a user interface (e.g., having navigation buttons, etc.) and the kiosk further acts as a filter to limit the URLs the client can traverse so that only HTML pages (URLs) defined by the kiosk builder are accessible.

Brief Summary Text - BSTX (22): An object of this invention is a server system and method that can deliver very large number of stored or created applications to reconfigure remote kiosks.

Brief Summary Text - BSTX (24): The present invention is a server system and method that is connected to one or more networks, e.g., the internet corporate or government intranets, etc. The server(s) serve application files (also called configuration sets) over the network(s) to one or more remote kiosks in order to configure and reconfigure the kiosks to perform various applications that are defined by the application files.

Detailed Description Text - DETX (8): An application is any use for which the kiosk is configured. For example, applications include uses (configurations) in the following fields: financial, business, information (news, advertising), communications (electronic mail, web access, video conferencing), retail, marketing, services (e.g., government programs). An application owner is any person, organization, or business that would configure the kiosk to provide the application. For example, a bank or mutual fund would configure the kiosk with one or more financial applications. Examples of these financial applications include providing the user with financial information, opening an account, dispensing cash, paying bills, applying for loans, making deposits, and obtaining assistance from the agent. An example of a service owner would be a car rental company that would configure the kiosk to provide a car rental/lease, etc.

Detailed Description Text - DETX (13): In an additional preferred configuration of the kiosk (100, 200), the server 195 provides the kiosk with application files 175 that are used to monitor or maintain the kiosk. For example, one or more of the embedded control programs 620 in these embodiments monitor the operating status of one or more of the input/output devices 130, e.g. by using "dead man" timer status, error checking protocols, etc. to determine which input/output devices are operational. This information is communicated back to the server 195. Other applications files 175 are used to query which input/output devices 130 are installed or operational in a given kiosk. In this way, the server 195 can determine which other application files 175 to send to the kiosk to enable the installed or operational input/output devices 130 and not to enable (configure) the uninstalled or faulty devices. Therefore, kiosks containing any general combination of input/output devices 130 can be installed remote to the server and the server will provide the correct and operational application files to make the kiosk operational for any given application. The application files can also be used to acquire information from one of more of the input/output devices to determine how to operate the devices.

Detailed Description Text - DETX (32): In one embodiment of this invention, parts 640A of kiosk specific control mechanisms 640 are added to the browser 160 and other parts 640B of the kiosk specific control mechanism 640 are added to the application programming interfaces (APIs) 680 (including 440) in order to enable the application files 175 to configure the kiosk. Accordingly, the kiosk specific control mechanisms 640 are divided into two parts: a browser mechanism 640A and an API mechanism 640B. In this embodiment, the browser mechanism 640A and the API mechanism 640B communicate through an interprocess communication (IPC) 640I. The IPC 640I interface allows the browser mechanism 640A and the API mechanism 640B to communicate using message passing instead direct function calls. (IPCs are well

known, one example would be the use of the Dynamic Data Exchange (DDE) in the Microsoft Window's operating system.)

Detailed Description Text - DETX (65): 6. The user may select the email function. The screen will show the HTML application for email. The embedded control program 620 may invoke related API function 640A or directly communicate with the mail server and directory server through the browser to identify the user and to retrieve existing email messages or sending new messages.

Detailed Description Text - DETX (104): In this embodiment, the browser mechanism 1040A is implemented by a plug-in technique (refer to "Programming Netscape Plug-ins" by Zan Oliphant, Sams.net Publishing, 1996 herein incorporated by reference in its entirety). The plug-in technique uses a native code module i.e., implemented using C or C++ or similar programming language, and in addition, in a more preferred embodiment, a Java wrapper. The plug-ins 1040A are located in a special plug-in directory specified by the browser 160. When the HTML interpreter 610 encounters the embedded file (620) that identifies the respective plug-in 1040A by a unique file name extension in the embedded file, also called Multipurpose Internet Mail Extension (MIME) type, the plug-in 1040A is dynamically loaded into the browser 160.

Claims Text - CLTX (17): 15. A server, as in claim 14, where the communication application includes any one or more of the following: a telephone call, an electronic mailing, a teleconference, a fax transmission, a training session, a search for information on the network, and a web based collaboration.

DIALOG 02 FEBRUARY 2005

File 2:INSPEC 1969-2005/Jan W4 (c) 2005 Institution of Electrical Engineers
File 9:Business & Industry(R) Jul/1994-2005/Feb 01 (c) 2005 The Gale Group
File 15:ABI/Inform(R) 1971-2005/Feb 01 (c) 2005 ProQuest Info&Learning
File 16:Gale Group PROMT(R) 1990-2005/Feb 02 (c) 2005 The Gale Group
File 20:Dialog Global Reporter 1997-2005/Feb 02 (c) 2005 The Dialog Corp.
File 35:Dissertation Abs Online 1861-2005/Jan (c) 2005 ProQuest Info&Learning
File 65:Inside Conferences 1993-2005/Jan W5 (c) 2005 BLDSC all rts. reserv.
File 99:Wilson Appl. Sci & Tech Abs 1983-2004/Nov (c) 2004 The HW Wilson Co.
File 148:Gale Group Trade & Industry DB 1976-2005/Feb 01 (c)2005 The Gale Group
File 160:Gale Group PROMT(R) 1972-1989 (c) 1999 The Gale Group
File 256:TecInfoSource 82-2004/Dec (c) 2004 Info.Sources Inc
File 275:Gale Group Computer DB(TM) 1983-2005/Feb 02 (c) 2005 The Gale Group
File 347:JAPIO Nov 1976-2004/Aug(Updated 041203) (c) 2004 JPO & JAPIO
File 348:EUROPEAN PATENTS 1978-2005/Jan W03 (c) 2005 European Patent Office
File 349:PCT FULLTEXT 1979-2002/UB=20050127,UT=20050120 (c) 2005 WIPO/Univentio
File 474:New York Times Abs 1969-2005/Feb 01 (c) 2005 The New York Times
File 475:Wall Street Journal Abs 1973-2005/Feb 01 (c) 2005 The New York Times
File 476:Financial Times Fulltext 1982-2005/Feb 02 (c) 2005 Financial Times Ltd
File 583:Gale Group Globalbase(TM) 1986-2002/Dec 13 (c) 2002 The Gale Group
File 610:Business Wire 1999-2005/Feb 02 (c) 2005 Business Wire.
File 613:PR Newswire 1999-2005/Feb 02 (c) 2005 PR Newswire Association Inc
File 621:Gale Group New Prod.Annou.(R) 1985-2005/Feb 02 (c) 2005 The Gale Group
File 624:McGraw-Hill Publications 1985-2005/Feb 01 (c) 2005 McGraw-Hill Co. Inc
File 634:San Jose Mercury Jun 1985-2005/Feb 01 (c) 2005 San Jose Mercury News
File 636:Gale Group Newsletter DB(TM) 1987-2005/Feb 02 (c) 2005 The Gale Group
File 810:Business Wire 1986-1999/Feb 28 (c) 1999 Business Wire
File 813:PR Newswire 1987-1999/Apr 30 (c) 1999 PR Newswire Association Inc

Set	Items	Description
S1	1193124	(CONFIGUR?????? OR INSTALL?????? OR INITIALIZ?????? OR LOAD??? OR DOWNLOAD???) (5N) (SOFTWARE OR APPLICATION OR PROGRAM OR FEATURE OR DEVICE OR COMPUTER)
S2	22290	(RECONFIGUR?????? OR REINSTALL?????? OR REINITIALIZ?????? OR RELOAD???) (5N) (SOFTWARE OR APPLICATION OR PROGRAM OR FEATURE OR DEVICE OR COMPUTERS)
S3	33867	(S1 OR S2) (5N) (REMOTE OR REMOTELY OR CENTER OR CENTRAL OR CENTRALLY)
S4	6868	S3 (5N) (COMMUNICATION OR LINE OR LINK OR CHANNEL OR WEB OR WWW OR NET OR LAN OR WAN OR INTERNET OR NETWORK)
S5	28509	(CONFIGUR?????? OR INSTALL?????? OR INITIALIZ?????? OR LOAD??? OR DOWNLOAD???) (5N) (AUTHORIZ?????? OR AUTHENTICAT???? OR VERIFY OR VERIFIED)
S6	194976	(CONFIGUR?????? OR INSTALL?????? OR INITIALIZ?????? OR LOAD??? OR DOWNLOAD???) (5N) (VERIFYING OR VERIFICATION OR ENABL??? OR PERMIT???? OR PERMITTED OR PERMISSION)
S7	549	(RECONFIGUR?????? OR REINSTALL?????? OR REINITIALIZ?????? OR RELOAD???) (5N) (AUTHORIZ?????? OR AUTHENTICAT???? OR VERIFY OR VERIFIED)
S8	6130	(RECONFIGUR?????? OR REINSTALL?????? OR REINITIALIZ?????? OR RELOAD???) (5N) (VERIFYING OR VERIFICATION OR ENABL??? OR PERMIT???? OR PERMITTED OR PERMISSION)
S9	1143	S3 AND S4 AND (S5 OR S6 OR S7 OR S8)
S10	30421	(S1 OR S2) (10N) (S5 OR S6 OR S7 OR S8)
S11	616	S9 AND S10
S12	294	S11 AND (FRANK OR FRANKING OR MAIL OR MAILING OR SHIP OR SHIPPING OR POSTAGE OR METER OR METERING)
S13	213	RD S12 (unique items) [Scanned ti,pd,kwic all]